

MODULE 4 : DES SAUTS ET DES CONSTANTES

Tu as créé un axe vertical dans ton monde virtuel. Ce serait donc intéressant de pouvoir apprendre au joueur comment « sauter ». Quand on dit « sauter », on veut dire la possibilité pour le joueur de se déplacer vers le haut et vers le bas.

SAUTER

Pour commencer, on aura besoin de modifier le code pour dire au joueur de sauter. On devra aussi déterminer quelle direction est « vers le haut ».

1. Clique sur l'icône du script à côté de « Player ».
2. Ajoute ceci sous la ligne 1 « extends KinematicBody2D » :
const UP = Vector2(0,-1)
 - a. Cela permet de définir la direction « vers le haut ».
3. Un peu plus bas, sous les lignes « else » et « motion », on devra ajouter l'action de sauter :

If.is_on_floor():

If Input.is_action_just_pressed("ui_up"):

Motion.y = -400

4. On va ajouter notre constante à tous les mouvements, donc remplace « move_and_slide(motion) » par :

move_and_slide(motion,UP)

Après avoir ajouté ces nouvelles lignes, ton code devrait ressembler à ceci. Enregistre-le et teste-le!

```

2
3  const UP = Vector2(0,-1)
4
5  var motion = Vector2()
6  |
7  func _physics_process(delta):
8  >|  motion.y+=10
9  >|  if Input.is_action_pressed("ui_right"):
10 >| >|  motion.x = 100
11 >| >|
12 >| elif Input.is_action_pressed("ui_left"):
13 >| >|  motion.x=-100
14 >| >|
15 >| else:
16 >| >|  motion.x=0
17 >| >|  if.is_on_floor():
18 >| >| >|  if Input.is_action_just_pressed("ui_up"):
19 >| >| >| >|  motion.y= -400
20 >| >| >|  move_and_slide(motion,UP)
21 >| >| >|  pass

```

CONSEIL DE PRO :

- Qu'est-ce qu'une constante? Comme leur nom l'indique, les constantes sont constantes, ce qui signifie qu'au lieu d'introduire des variables différentes dans tout le code, elles permettent d'utiliser partout les mêmes variables.
- Les constantes s'écrivent en lettres majuscules.
- Exemple : au lieu de « -100 » pour le mouvement, tu peux le remplacer par « -SPEED ». De cette manière, tu n'auras besoin de le changer qu'une fois, plutôt que de le modifier chaque fois que tu l'ajoutes dans le code!

CRÉER DES CONSTANTES

Ajoutons une constante à notre code! Remplaçons « -100 » pour le mouvement (« motion », en anglais) par « -SPEED ».

1. Clique sur l'icône du script à côté de « Player ».
2. On va ajouter d'autres constantes après « const UP ».

const SPEED = 100

const GRAVITY = 10

const JUMP_HEIGHT = -400

3. Remplace toutes tes variables par ces constantes.
 - a. Exemple : motion.x = 100 → motion.x = SPEED

Après avoir ajouté ces nouvelles lignes, ton code devrait maintenant ressembler à ceci. Enregistre-le et teste-le! Il devrait fonctionner de la même façon, mais les constantes rendent le codage plus facile (par exemple, plus besoin de taper « 100 » à chaque fois, tu peux taper « SPEED » à la place).

```

1 extends KinematicBody2D
2
3 const UP = Vector2(0,-1)
4 const SPEED = 200
5 const GRAVITY = 10
6 const JUMP_HEIGHT = -400
7
8 var motion = Vector2()
9
10 func _physics_process(delta):
11     motion.y+=GRAVITY
12     if Input.is_action_pressed("ui_right"):
13         motion.x = SPEED
14
15     elif Input.is_action_pressed("ui_left"):
16         motion.x = -SPEED
17
18     else:
19         motion.x=0
20     if is_on_floor():
21         if Input.is_action_just_pressed("ui_up"):
22             motion.y = JUMP_HEIGHT
23         move_and_slide(motion,UP)
24     pass
25

```

CODE FINAL

Voilà à quoi devrait ressembler ton code final :

```

1  extends KinematicBody2D
2
3  const UP = Vector2(0,-1)
4  const SPEED = 200
5  const GRAVITY = 10
6  const JUMP_HEIGHT = -400
7
8  var motion = Vector2()
9
10 func _physics_process(delta):
11     motion.y+=GRAVITY
12     if Input.is_action_pressed("ui_right"):
13         motion.x = SPEED
14
15     elif Input.is_action_pressed("ui_left"):
16         motion.x = -SPEED
17
18     else:
19         motion.x=0
20     if.is_on_floor():
21         if Input.is_action_just_pressed("ui_up"):
22             motion.y = JUMP_HEIGHT
23     move_and_slide(motion,UP)
24     pass
25

```

VOYONS MAINTENANT CE QUE TU PEUX FAIRE!

Avec moins de 25 lignes de code, tu as déjà réussi à créer un jeu vidéo! Il est bien sûr très basique mais à ton avis, qu'est-ce que tu pourrais programmer d'autre comme action pour ton joueur? Il y a aussi beaucoup d'autres éléments que tu peux ajouter à ton jeu en 2D, comme de l'animation ou des lumières et des ombres. Quand tu maîtriseras tout ça, tu pourras passer aux outils de Godot pour créer des jeux en 3D super réalistes!

Godot utilise GDScript comme langage de programmation. Mais en te familiarisant avec son API (« Application Programming Interface » en anglais ou « interface de programmation » en français), tu apprendras énormément de principes de programmation qui te seront utiles dans ton parcours de développement de jeux ou Web! Approfondis tes connaissances en apprenant comment fonctionnent C#, C++ ou n'importe quel autre langage de programmation populaire!

[Découvrir plus de modules pour Développement de jeu avec Godot!](#)

[Découvrir le tutoriel vidéo sur Godot!](#)

[Retourner à l'Apprentissage en ligne Académie Geek Squad!](#)

[Suggestions, questions ou commentaires? Envoyez-nous un courriel à \[academy@geeksgquad.ca\]\(mailto:academy@geeksgquad.ca\)](mailto:academy@geeksgquad.ca)